THREE DIMENSIONAL TIME SPLITTING TECHNIQUE: CONSTRUCTION OF MIXED HYBRID FINITE ELEMENTS AND HIGH RESOLUTION FINITE VOLUME SCHEMES ON TETRAHEDRAL MESH

Annamaria Mazzia

Padova, Marzo 2002 Rapporto Tecnico N° 86

Abstract

Godunov Mixed Methods on triangular grids has been shown to be an effective tool for the solution of the two-dimensional advection-dispersion equation. The method is based on the discretization of the dispersive flux by means of the mixed hybrid finite element approach, while a high resolution Godunov-like finite volume scheme discretizes advection. The two techniques are combined together through a time-splitting algorithm that achieves formal second order accuracy if a corrective term is added in the finite volume stencil. In this paper we develop and study the extension of this approach to three dimensions employing tetrahedral elements and a fully 3D limiter. Particular attention is devoted to the choice of a truly three-dimensional limiter that preserves second order accuracy in space. To this aim, several generalizations of two dimensional finite volumes schemes are presented and their behavior in three dimensions is analyzed. The numerical characteristics of the proposed method is studied both theoretically and numerically using simple test problems.

1 Introduction

In this paper we propose a time-splitting approach for the numerical solution of the advection-dispersion equation in three dimensions, on three-dimensional triangulations. In two dimensions, this technique has been developed as a reliable tool for producing accurate numerical approximations of advection-dispersion equations, discretizing indipendently the advective and dispersive terms with techniques deemed most appropriate [4, 11]. High resolution Godunov-type Finite Volume (HRFV) schemes for discretizing advection have been employed in combination with a Mixed Hybrid Finite Elements (MHFE) scheme for the discretization of the dispersion term [11, 12, 10, 13]. The extension of this technique to three dimensions requires careful consideration in particular on the adoption of an appropriate HRFV scheme combined with a truly three-dimensional limiter that preserves second order accuracy in space.

Extensions of cell centered Finite Volume (FV) schemes for unstructured triangular meshes to tetrahedra seem to be readily feasible from a theoretical point of view. Practically, when generalizing these schemes to three dimensional triangulations, application

of a truly three-dimensional limiter for modifying the underlying finite volume schemes in such a way that the resulting method is monotone can drastically reduce the order of accuracy, yielding poor results. The reason for this behavior is mainly attributed to the poorly structured tetrahedra that need to be used in a 3D mesh. Methods as proposed in [5, 2, 7] that work well in two dimensions, generalized to tetrahedra result in a first order accurate scheme. The limiters proposed to enforce a positivity constraint on these schemes may be not appropriate when extended to tetrahedra. The great computational effort to choose the right interpolant for the linear reconstruction (four linear interpolant in the case of the extension of Durlofsky et al. [5] scheme and five interpolant in the case of Batten et al. scheme [2]) does not correspond to an improvement in terms of accuracy of the solution. Even if it has been noted that the use of linear reconstructions does not always imply second or higher order accuracy even when the solution is smooth [1], such failure of the above generalized methods requires more investigation.

On the contrary, other schemes seem to be quite suited to be applied on tetrahedra. Indeed, better results are obtained when applying the so called extremum limiter, proposed by [15], and the limiter introduced by Barth and Jespersen [1]. Application of the above limiters guarantees that the reconstructed values along the faces of each tetrahedron well satisfy a local maximum principle. Anyway, the numerical results show that more confidence can be given to the Barth-Jespersen limiter.

The latter strategy in combination with a linear reconstruction obtained by the least squares method (LSM) is then applied to the time splitting technique.

The time splitting technique that combines the HRFV scheme based on the LSM method for advection with the MHFE scheme for dispersion preserves the same accuracy, robustness and convergence properties studied and verified in the two dimensional case. Simple examples of one and two dimensional problems solved on tetrahedral meshes confirm the theoretical behavior of the proposed scheme.

This paper is organized as follows: in Section 2 we introduce the advection-dispersion equation governing subsurface contaminant transport and give an overview of the time-splitting technique. In Section 3 we report on the construction of the three dimensional MHFE method and describe several computational details. In Section 4 we state an outline of HRFV schemes with different linear reconstructions and limiters. In Section 5 we discuss

the numerical results obtained for simple advection, dispersion and advection-dispersion problems. Finally, in Section 7 we propose some conclusions arising from the work. For the sake of completeness, in appendix are included some computational notes useful when working in two dimensions. Practically they recall some details described in Sections 3 and 4.

2 Time splitting algorithm

In the following, we refer to the advection-dispersion equation governing subsurface contaminant transport:

$$\begin{aligned}
&\frac{\partial \phi c}{\partial t} + \vec{\nabla} \cdot (\vec{v}c - D\vec{\nabla}c) = f && \text{on } \Omega \times (0,T], \\
&c = c^0 && \text{on } \Omega \times 0, \\
&c = b_D && \text{on } \Gamma_D \times (0,T], \\
&-D\vec{\nabla}c \cdot \vec{n} = b_N && \text{on } \Gamma_N \times (0,T] \\
&(\vec{v}c - D\vec{\nabla}c) \cdot \vec{n} = b_C && \text{on } \Gamma_C \times (0,T]
\end{aligned} \tag{2.1}$$

where c is the concentration of the solute, $\phi(t)$ is the porosity of the medium, t is time $\vec{v} = \vec{v}(\vec{P},t)$ is Darcy's velocity, $\vec{P} = (x,y,z)$ is the Cartesian spatial coordinate vector, $D = D(\vec{v})$ is the tensor accounting for mechanical dispersion and molecular diffusion, and f is a source or sink term. Moreover, $\vec{n}$ is the outward normal unit vector, c^o is concentration at time 0, b_D is the prescribed concentration (Dirichlet condition) on boundary Γ_D, b_N is the prescribed dispersive flux (Neumann condition) across boundary Γ_N, b_C is the prescribed total flulx of solute (Cauchy conditions) across boundary Γ_C ($\Omega \in \mathbb{R}^3$, $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_C = \partial\Omega$ and T is the final time of observation). In the following, we set $\phi = 1$, since its variation with time is neglected in the transport equation, being in general much smaller than the concentration variations.

Denoting by $\vec{F}$ and $\vec{G}$ the advective and dispersive fluxes, respectively, equation (2.1) may be written as:

$$\frac{\partial c}{\partial t} + \vec{\nabla} \cdot (\vec{F} + \vec{G}) = f \qquad \text{on } \Omega \times (0,T] \tag{2.2}$$

$$\vec{F} = \vec{v}c \tag{2.3}$$

$$\vec{G} = -D\vec{\nabla}c \tag{2.4}$$

The aim of the time splitting algorithm is to solve equations (2.2) taking into account the splitting of the advection and the dispersion fluxes into two separate partial differential equations, respectively.

The time splitting technique, applied on a three dimensional mesh, discretizes the domain Ω into m tetrahedra T_l, $l = 1, \ldots, m$. Concentration c is then approximated by:

$$c \simeq \tilde{c} = \sum_{l=1}^{m} c_l \psi_l,$$

where ψ_l are $P_0(T_l)$ scalar basis functions, taking on the value one on tetrahedron T_l and zero elsewhere.

The advection-dispersion equation (2.2) is discretized using the time splitting algorithm that combines a HRFV scheme for the advection term and a MHFE scheme for the dispersion term. This technique can be viewed as a predictor-corrector approach as follows. Multiplying equation (2.2) by ψ_l and integrating in space and time, with time step Δt over the time interval $[t^k, t^{k+1}]$, the following semidiscrete equations are obtained:

$$c_l^{k+1} = c_l^k - \frac{\Delta t}{|T_l|} \int_{T_l} [\vec{\nabla} \cdot (\vec{F}(c^{k+1-\theta}) + \vec{G}(c^{k+\theta})) - f^{k+\theta}] \, \mathrm{d}\Delta, \qquad l = 1, \ldots, m$$

where c_l^k is the volume average over T_l defined by $c_l^k = \int_{T_l} c(\cdot, t^k) \, \mathrm{d}\Delta / |T_l|$, $|T_l|$ is the volume of T_l and a weighted scheme is used for the time quadrature with weighting parameter $\theta \in [0.5, 1]$ and $c^{k+\theta} = \theta c^{k+1} + (1 - \theta) c^k$.

Integration in time is explicit for the FV scheme and implicit for the MHFE method. We combine explicit and implicit first order Euler scheme ($\theta = 1$) or explicit and implicit second order mid-point rule ($\theta = 0.5$). Since the stability of the advection discretization is determined by the CFL constraint, while the dispersive step is not subject to stability restrictions, different time steps sizes are allowed for advection and dispersion, Δt_a and Δt_d, respectively. Thus a finer advection time step together with a coarser dispersive time step can be employed. The relationship between the two time steps is: $\Delta t_d = n_a \Delta t_a$, with n_a integer ≥ 1.

The accuracy of the scheme is influenced by the accuracy of the two spatial discretization methods employed. Namely, second order in space is obtained at the centroids of the tetrahedra, as both HRFV and MHFE are there spatially second order accurate. The time-splitting algorithm, however, introduces an $O(\Delta t)$ error in the overall procedure, which is

thus only first order accurate in time [11]. Second order accuracy in space and time can be achieved with $n_a = 1$ when a correction term is added to the FV scheme [10]. It is straightforward to choose $n_a \geq 1$ when first order schemes in time are employed, and $n_a = 1$ otherwise, to avoid order reduction. The correction term is required to take into account the influence of dispersion when applying the midpoint rule in the FV scheme and is simply obtained adding a backward finite difference in the linear reconstruction of the FV discretization.

Denoting by L_a and L_d the advective and dispersive numerical fluxes, respectively, the time splitting technique can be summarized in the following way [11, 12]:

ALGORITHM 2.1.

For each time step do:

- *advection step: for each T_l solve n_a times with the explicit FV scheme ($n_a \geq 1$ when $\theta = 1$, $n_a = 1$ otherwise), using Δt_a as the time step, determining the predictor concentration $\hat{c}_l^{k+1}$*

  1. $c_l^{(0)} := c_l^k$
  2. DO $i_a = 0, n_a - 1$

$$c_l^{(i_a+1)} = c_l^{(i_a)} + \Delta t_a \left[L_a(c_l^{(i_a+1-\theta)}) \right] \tag{2.5}$$

     END DO

  3. $\hat{c}_l^{k+1} := c_l^{(n_a)}$

- *dispersion step: for each T_l solve with implicit MHFE method using $\hat{c}_l^{k+1}$ as initial condition*

$$c_l^{k+1} = \hat{c}_l^{k+1} + \Delta t_d \left[L_d(c_l^{k+\theta}) \right] \tag{2.6}$$

  with $\Delta t_d = \Delta t = n_a \Delta t_a$, obtaining the final approximation c_l^{k+1}.

3 MHFE discretization

The MHFE formulation applied to the discretization of the dispersive step extends to tetrahedra the algorithm applied to triangles in the two dimensional case [9, 11]. Given the

tetrahedral mesh, the dispersive flux $\vec{G}$ is approximated on tetrahedron T_l as $\vec{G} \approx \sum_{j=1}^{4} g_{jl}\vec{w}_{jl}$ where $\vec{w}_{jl}$ are the discontinous RT0 vector basis functions. The MHFE formulation introduces the unknown Lagrange multiplier expressed as $\lambda = \sum_{j=1}^{n} \lambda_j \mu_j$, where n is the number of faces of the mesh, λ_j represents the trace of the concentration on the face f_j and μ_j are piecewise constant basis functions.

Set the time step Δt_d (Δt) on the time interval $[t^k, t^{k+1}]$, discretization of equation (2.6) by MHFE in space and the θ-weighted scheme in time yields:

$$\begin{pmatrix} A & -B & Q \\ B^T & P' & 0 \\ Q^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{g}^{k+\theta} \\ \mathbf{c}^{k+\theta} \\ \boldsymbol{\lambda}^{k+\theta} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{f} + \tilde{P}\hat{\mathbf{c}}^{k+1} \\ \mathbf{b}_N \end{pmatrix} \tag{3.1}$$

where $A = \mathrm{diag}[A_1, \ldots, A_m]$, $B = \mathrm{diag}[B_1, \ldots, B_m]$ and

$$\begin{aligned} A_l &= (a_{ik}) = \int_{T_l} D_l^{-1} \vec{w}_{il} \cdot \vec{w}_{kl} \, \mathrm{d}\Delta \\ B_l &= (b_i) = \int_{T_l} \vec{\nabla} \cdot \vec{w}_{il} \, \mathrm{d}\Delta \\ Q &= (q_{rj}) = \int_{\partial T_l} \mu_j \vec{w}_{il} \cdot \vec{n}_l \, \mathrm{d}\Gamma \\ \mathbf{g} &= (g_r) = g_{il} \\ \mathbf{f} &= (\hat{f}_l) = \int_{T_l} f_l \, \mathrm{d}\Delta \end{aligned}$$

where $i, k = 1, 2, 3, 4$, $j = 1, \ldots, n$, $r = 4(l-1) + i$ and $\mathbf{c} = (c_l)$, $\boldsymbol{\lambda} = (\lambda_j)$, and $\mathbf{b}_N = (b_{Nj})$ where b_{Nj} assumes a non vanishing value only if there is a Neumann condition on a boundary face f_j. Setting $P^k = \mathrm{diag}[p_1^k, \ldots, p_m^k]$ with $p_l^k = \phi_l^k |T_l| / \Delta t$ then $P' = \dfrac{P^{k+1}}{\theta}$ and $\tilde{P} = P' - P^{k+1} + P^k$. If p_l^k is not dependent on time, then $P' = \tilde{P}$. The vector $\vec{n}_l$ represents the outward normal to T_l. Along each face of e_j T_l it is expressed as $\vec{n}_{jl}$.

A final system of equations for the unknown λ is obtained:

$$Q^T M Q \boldsymbol{\lambda}^{k+\theta} = Q^T S H^{-1} (\mathbf{f} + \tilde{P}\hat{\mathbf{c}}^{k+1}) - \mathbf{b}_N \tag{3.2}$$

where $H = D' + B^T A^{-1} B$, $S = A^{-1} B$, and $M = A^{-1} - S H^{-1} S^T$. The system is symmetric and positive and can be solved efficiently by the Preconditioned Conjugate Gradient method. Once $\boldsymbol{\lambda}^{k+\theta}$ is calculated, $\mathbf{c}^{k+1}$ is easily evaluated.

3.1 Details about integrals computation

Computation of the entries a_{ik}, b_i, q_{rj} of system (3.1) requires knowledge of some geometrical relationships and of the RT0 basis functions on tetrahedra. In the following we will consider a generic tetrahedron T (or T_l), with faces e_1, e_2, e_3, e_4 ($e_{1l}, e_{2l}, e_{3l}, e_{4l}$, respectively) and vertices (or nodes) $\vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{P}_4$.

3.1.1 Surface through $\vec{P}_1, \vec{P}_2, \vec{P}_3$

Given a face e_i with vertices $\vec{P}_j = (x_j, y_j, z_j)$, $j = 1, 2, 3$, the surface passing through $\vec{P}_1$, $\vec{P}_2$, $\vec{P}_3$ can be written as:

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0 \tag{3.3}$$

Equation (3.3) is equivalent to

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 & 0 \\ x_1 & y_1 & z_1 & 1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 & 0 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 & 0 \end{vmatrix} = 0 \Leftrightarrow \begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0 \Leftrightarrow$$

$$\Leftrightarrow \alpha(x - x_1) + \beta(y - y_1) + \gamma(z - z_1) = 0 \tag{3.4}$$

where

$$\alpha = \begin{vmatrix} y_2 - y_1 & z_2 - z_1 \\ y_3 - y_1 & z_3 - z_1 \end{vmatrix} = (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1),$$

$$\beta = -\begin{vmatrix} x_2 - x_1 & z_2 - z_1 \\ x_3 - x_1 & z_3 - z_1 \end{vmatrix} = (x_3 - x_1)(z_2 - z_1) - (x_2 - x_1)(z_3 - z_1),$$

$$\gamma = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1).$$

Alternatevely, α, β, γ can also be written as:

$$\alpha = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}, \ \beta = -\begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{vmatrix}, \ \gamma = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}.$$

LEMMA 3.1. *It is straightforward to prove that*

$$x_1\alpha + y_1\beta + z_1\gamma = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \tag{3.5}$$

Proof. A first way to prove (3.5) is by direct substitution of the values defining α, β and γ.

A second way is given by writing the surface (3.3) through $\vec{P}_1, \vec{P}_2, \vec{P}_3$ as

$$x\begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix} - y\begin{vmatrix} x_1 & z_1 & 1 \\ x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \end{vmatrix} + z\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0 \Leftrightarrow$$

$$\Leftrightarrow \alpha x + \beta y + \gamma z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0 \tag{3.6}$$

Since the same surface can be written as (3.4), by comparison we obtain relation (3.5).

□

3.1.2 Normal to the surface

Introducing $\vec{n}$, the normal to the surface, that is to a face of the tetrahedron, its normalized components can be written as:

$$n_x = \frac{\alpha}{\sqrt{\alpha^2+\beta^2+\gamma^2}},\ n_y = \frac{\beta}{\sqrt{\alpha^2+\beta^2+\gamma^2}},\ n_z = \frac{\gamma}{\sqrt{\alpha^2+\beta^2+\gamma^2}}.$$

Therefore the surface equation (3.4) becomes:

$$n_x(x-x_1) + n_y(y-y_1) + n_z(z-z_1) = 0 \tag{3.7}$$

The normal $\vec{n}$ must be external to the tetrahedron. At the moment, we consider $\vec{n}$ as external, next we will see the difference between internal and external normal and how to choose the right one.

3.1.3 Integrals computation

An other useful consideration is concerning the computation of a surface integral. Given the surface $S = S(x, y, z)$ such that $z = f(x, y)$, z depending on x and y, with $(x, y) \in D$, D being a 2D section of S, then

$$\int_S g(x, y, z)\ \mathrm{d}S = \int_D g(x, y, f(x, y))\sqrt{1 + f_x^2 + f_y^2}\ \mathrm{d}x\ \mathrm{d}y.$$

In the case of $S = e_i$, from equation (3.7) we obtain

$$z = z_1 - \frac{n_x}{n_z}(x - x_1) - \frac{n_y}{n_z}(y - y_1)$$

and

$$\int_{e_i} g(x, y, z)\ \mathrm{d}S = \int\int_D g(x, y, z_1 - \frac{n_x}{n_z}(x - x_1) - \frac{n_y}{n_z}(y - y_1))\frac{1}{|n_z|}\ \mathrm{d}x\ \mathrm{d}y. \tag{3.8}$$

3.1.4 RT0 basis functions

For each tetrahedron T_l, $l = 1, \ldots, m$, with faces $e_{1l}, e_{2l}, e_{3l}, e_{4l}$, the RT0 basis functions $\vec{w}_{il}$ $(i = 1, 2, 3, 4)$ are of the following form:

$$\vec{w}_{il} = \begin{pmatrix} a_{il}x + b_{il} \\ a_{il}y + c_{il} \\ a_{il}z + d_{il} \end{pmatrix}$$

and satisfy the following property:

$$\int_{e_{il}} \vec{w}_{jl} \cdot \vec{n}_{il}\ \mathrm{d}S = \delta_{ij} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

where $\vec{n}_{il}$ is the outward normal to e_{il}. For sake of simplicity, we will omit the subscripts il when referring to the coefficients a, b, c, d.

Taking into account relation (3.8), computation of the integral $\int_{e_i} \vec{w}_{jl} \cdot \vec{n}_{il}\ \mathrm{d}S$ is given in the following way (we consider a face with nodes (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)):

$$\begin{aligned}
&\int_{e_i} \vec{w}_{jl} \cdot \vec{n}_{il}\ \mathrm{d}S = \int_{e_i} (ax + b)n_x + (ay + c)n_y + (az + d)n_z\ \mathrm{d}S = \\
&a\int_D \{xn_x + yn_y + [z_1 - \frac{n_x}{n_z}(x - x_1) - \frac{n_y}{n_z}(y - y_1)]n_z\}\frac{1}{|n_z|}\ \mathrm{d}x\ \mathrm{d}y + \int_D (bn_x + cn_y + dn_z)\frac{1}{|n_z|}dxdy = \\
&a\int_D (x_1n_x + y_1n_y + z_1n_z)\frac{1}{|n_z|}\ \mathrm{d}x\ \mathrm{d}y + \int_D (bn_x + cn_y + dn_z)\frac{1}{|n_z|}\ \mathrm{d}x\ \mathrm{d}y = \\
&a(x_1n_x + y_1n_y + z_1n_z)\frac{|D|}{|n_z|} + (bn_x + cn_y + dn_z)\frac{|D|}{|n_z|}
\end{aligned}$$

TABLE 3.1: *Vertices and opposite node of each face.*

face	vertices			opposite
e_1	$\vec{P}_1$	$\vec{P}_2$	$\vec{P}_3$	$\vec{P}_4$
e_2	$\vec{P}_2$	$\vec{P}_3$	$\vec{P}_4$	$\vec{P}_1$
e_3	$\vec{P}_3$	$\vec{P}_4$	$\vec{P}_1$	$\vec{P}_2$
e_4	$\vec{P}_4$	$\vec{P}_1$	$\vec{P}_2$	$\vec{P}_3$

where $|D|$ is the area of the triangle with vertices (x_1, y_1), (x_2, y_2), (x_3, y_3), that is the absolute value of

$$\frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2}\gamma$$

Therefore $\dfrac{|D|}{|n_z|} = \dfrac{1}{2}\sqrt{\alpha^2+\beta^2+\gamma^2}$ and integral $\displaystyle\int_{e_i} \vec{w}_{jl} \cdot \vec{n}_{il}\ \mathrm{d}S$ becomes:

$$\int_{e_i} \vec{w}_{jl} \cdot \vec{n}_{il}\ \mathrm{d}S = \frac{a}{2}(x_1\alpha + y_1\beta + z_1\gamma) + \frac{1}{2}(b\alpha + c\beta + d\gamma)$$

Let us consider a generic tetrahedron T with faces and vertices as described in Table 3.1. Therefore, taking into account, for example, the face e_2 opposite to node P_1, we can write:

$$\int_{e_2} \vec{w}_j \cdot \vec{n}_2\ \mathrm{d} = \frac{a}{2}(x_2\alpha + y_2\beta + z_2\gamma) + \frac{1}{2}(b\alpha + c\beta + d\gamma)$$

where α, β, γ are the components of the normal vector to the face e_2. Note that now we have $x_2,\ y_2$ and z_2 in the right hand side of the previous relation.

From the literature, it is known that the classical 3D Galerkin function

$$N_1 = \frac{a_1 + b_1 x + c_1 y + d_1 z}{6V}$$

is such that its gradient is normal to the face e_2. The gradient is: $\vec{\nabla} N_1 = \dfrac{1}{6V}(b_1, c_1, d_1)$, with V the volume of the tetrahedron. Therefore, with reference to face e_2, we can set:

$$\alpha = b_1,\ \beta = c_1,\ \gamma = d_1.$$

Moreover, $\vec{\nabla} N_i$ is normal to face e_{i+1}, $i = 1, 2, 3$, while $\vec{\nabla} N_4$ is normal to e_1. In particular, the coefficients a_i, b_i, c_i, d_i, $i = 1, 2, 3, 4$ of the Galerkin functions are as described

in the following:

$$a_1 = \begin{vmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} \quad a_2 = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} \quad a_3 = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_4 & y_4 & z_4 \end{vmatrix} \quad a_4 = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

$$b_1 = -\begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix} \quad b_2 = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix} \quad b_3 = -\begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_4 & z_4 \end{vmatrix} \quad b_4 = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}$$

$$c_1 = \begin{vmatrix} 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{vmatrix} \quad c_2 = -\begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{vmatrix} \quad c_3 = \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_4 & z_4 \end{vmatrix} \quad c_4 = -\begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{vmatrix}$$

$$d_1 = -\begin{vmatrix} 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix} \quad d_2 = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix} \quad d_3 = -\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_4 & y_4 \end{vmatrix} \quad d_4 = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

Recalling relation (3.6) and substituting the values of b_4, c_4, d_4, a_4, the surface through $\vec{P}_1, \vec{P}_2, \vec{P}_3$ can be written as:

$$b_4 x + c_4 y + d_4 z + a_4 = 0$$

From comparison with equation (3.4) we obtain:

$$\alpha x_1 + \beta y_1 + \gamma z_1 = -a_4, \;\; \alpha = b_4, \;\; \beta = c_4, \;\; \gamma = d_4$$

Thus

$$\int_{e_1} \vec{w}_j \cdot \vec{n}_1 \, \mathrm{d}S = -\frac{a}{2} a_4 + \frac{1}{2}(b_4 b + c_4 c + d_4 d). \tag{3.10}$$

The unknowns are a, b, c, d of $\vec{w}_j$. Setting $j = 1$, from relation (3.9) we obtain:

$$\int_{e_1} \vec{w}_1 \cdot \vec{n_1} \, \mathrm{d}S = 1, \; \int_{e_2} \vec{w}_1 \cdot \vec{n_2} \, \mathrm{d}S = 0, \; \int_{e_3} \vec{w}_1 \cdot \vec{n_3} \, \mathrm{d}S = 0, \; \int_{e_4} \vec{w}_1 \cdot \vec{n_4} \, \mathrm{d}S = 0.$$

Substituting (3.10) we get:

$$\begin{cases} -a_4 a + b_4 b + c_4 c + d_4 d = 2 \\ -a_1 a + b_1 b + c_1 c + d_1 d = 0 \\ -a_2 a + b_2 b + c_2 c + d_2 d = 0 \\ -a_3 a + b_3 b + c_3 c + d_3 d = 0 \end{cases} \tag{3.11}$$

Solution of system (3.11) gives the values of a, b, c, d of w_1. Similar relations hold for the other basis functions. It is interesting to observe that from the previous system, we obtain:

$$-\sum_{i=1}^{4} a_i a + \sum_{i=1}^{4} b_i b + \sum_{i=1}^{4} c_i c + \sum_{i=1}^{4} d_i d = 2 \tag{3.12}$$

Is it straightforward to prove that $\sum_{i=1}^{4} b_i = \sum_{i=1}^{4} c_i = \sum_{i=1}^{4} d_i = 0$. Indeed,

$$\sum_{i=1}^{4} b_i = -\begin{vmatrix} 1 & 1 & y_1 & z_1 \\ 1 & 1 & y_2 & z_2 \\ 1 & 1 & y_3 & z_3 \\ 1 & 1 & y_4 & z_4 \end{vmatrix} = 0$$

(similar considerations hold for the summations in c_i and d_i).

Instead, it is very simple to prove that

$$\sum_{i=1}^{4} a_i = 6V$$

where V is the volume of the tetrahedron defined as

$$V = \frac{1}{6}\begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}$$

Therefore equation (3.12) becomes:

$$-6Va = 2$$

giving a relationship between a and the volume V:

$$a = -\frac{1}{3V}. \tag{3.13}$$

Now we are able to distinguish between inner and outward normal. By the divergence theorem we get:

$$\int_T \vec{\nabla} \cdot \vec{w}_1 \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z = \int_{\partial T} \vec{w}_1 \cdot \vec{n} \, \mathrm{d}S = \sum_{i=1}^{4} \int_{e_i} \vec{w}_1 \cdot \vec{n}_i \, \mathrm{d}S = 1$$

But,

$$\int_T \vec{\nabla} \cdot \vec{w}_1 \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z = \int_T (\frac{\partial \vec{w}_1}{\partial x} + \frac{\partial \vec{w}_1}{\partial y} + \frac{\partial \vec{w}_1}{\partial z}) \, \mathrm{d}V = \int_T 3a \, \mathrm{d}V = 3a|V|$$

Thus

$$3a|V| = 1 \tag{3.14}$$

and $a = \dfrac{1}{3|V|}$ must be a positive value. Relation (3.13) between a and V is written with no consideration about the sign of a. We are now able to conclude that if computation of a from system (3.11) gives a negative value for a, we have worked with an inner normal. In this case, it will be sufficient to change the signs of a, b, c, d and to consider $-\vec{w}_1$. If a is positive, then the normal is external.

These results are in accordance with the following statement, useful to decide if the normal is external or not. Given $\vec{\nabla} N_1$ (the normal to the face e_2) and the barycentrum of the tetrahedron T, $\vec{G} = (x_G, y_G, z_G)$, we consider the vector passing through $\vec{G}$ and a vertex of the face, for example $\vec{P}_4$: $\vec{G}\vec{P}_4 = (x_4 - x_G, y_4 - y_G, z_4 - z_G)$. The inner product $\vec{G}\vec{P}_4 \cdot \vec{\nabla} N_1$ is equal to $b_1(x_4 - x_G) + c_1(y_4 - y_G) + d_1(z_4 - z_G)$. If $\vec{G}\vec{P}_4 \cdot \vec{\nabla} N_1 > 0$ then the normal is external, otherwise we are considering an inner normal.

From a computational point of view, relation (3.13) says that the coefficient a_{il} is equal along the four faces of tetrahedron T_l. Set a as the positive value $a = \dfrac{1}{3|V|}$, the coefficient b, c, d are readily computed taking into account the first equation of system (3.11) (substituting the value of a and regardless of the sign) and the definition of the volume of a tetrahedron:

On the one hand, we have

$$-a_4 \frac{1}{3|V|} + b_4 b + c_4 c + d_4 d = \pm 2 \tag{3.15}$$

On the other and

$$6V = a_4 + x_4 b_4 + y_4 c_4 + z_4 d_4$$

By comparison, we obtain for the basis $\vec{w}_1$:

$$b = \frac{-x_4}{3|V|}, \qquad c = \frac{-y_4}{3|V|}, \qquad d = \frac{-z_4}{3|V|}.$$

The sign of (3.15) will be positive if V is negative and negative otherwise.

At the same way, the coefficients of $\vec{w}_j, j = 2, 3, 4$ are computed as:

$$b_j = \frac{-x_{j-1}}{3|V|}, \qquad c_j = \frac{-y_{j-1}}{3|V|}, \qquad d_j = \frac{-z_{j-1}}{3|V|}.$$

It is clear, that computation of a, b, c, d parameters is now very easily and not expensive.

3.1.5 Computation of $A,\ B,\ Q$

With the above considerations and using the same notations of system (3.1), we are able to compute the values of matrices $A,\ B$ and Q.

The values of Q are $q_{rj} = \int_{\partial T_l} \mu_j \vec{w}_{il} \cdot \vec{n}_l \ \mathrm{d}\Gamma$. With simple calculations we get:

$$q_{rj} = \mu_j$$

The non vanishing entries of matrix A $a_{ik} = \int_{T_l} D_l^{-1} \vec{w}_{il} \cdot \vec{w}_{kl} \ \mathrm{d}\Delta$ are computed by applying the Gauss-Lobatto formula to the following development:

$$a_{ik} = \int_{T_l} \frac{(a_i x + b_i)(a_k x + b_k)}{D_{xx}} + \frac{(a_i y + c_i)(a_k y + c_k)}{D_{yy}} + \frac{(a_i y + d_i)(a_k y + d_k)}{D_{zz}} \ \mathrm{d}\Delta$$

In the previous formula, matrix D_l is a diagonal matrix with entries D_{xx}, D_{yy}, D_{zz}.

The diagonal entries of matrix B are equal to 1. Indeed

$$\int_{T_l} \vec{\nabla} \cdot \vec{w}_{il} \ \mathrm{d}\Delta = \int_{T_l} (\frac{\partial \vec{w}_{il}}{\partial x} + \frac{\partial \vec{w}_{il}}{\partial y} + \frac{\partial \vec{w}_{il}}{\partial z}) \ \mathrm{d}V = \int_{T_l} (a_i + a_i + a_i) \ \mathrm{d}V = 3a_i|V| = 1$$

We have used equality (3.14).

4 HRFV discretization

Equation (2.5) can be explicitly written as:

$$\phi_l^{k+1} c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|} \int_{T_l} \vec{\nabla} \cdot (\vec{F}(c_l^{k+1-\theta}) \ \mathrm{d}\Delta \tag{4.1}$$

with $\ell = 1, \ldots, m$, and is solved using as initial condition the solution calculated at the and of the previous time step. In the two dimensional formulation, the discretization of the equation corresponding to (4.1) was obtained by means of the Finite Volume scheme on unstructured triangular grid, as developed by [5] and then modified by [8]. From a theoretical point of view, this scheme can be readily generalized to tetrahedra. In practice,

we observed that extension of two dimensional limiters may drastically reduce the order of accuracy, yielding poor results. Only few limiters preserve second order accuracy, as we will see in the numerical results. The reason for this behavior is mainly attributed to the poorly structured tetrahedra that need to be used in a three dimensional mesh.

In the following, we introduce the FV formulation and describe several HRFV methods developed on triangular grids in order to guarantee satisfaction of appropriate maximum principle. We directly present their generalization to three dimensions and the numerical results will be useful to decide the more suitable to be used in the time splitting technique.

Application of the divergence theorem to the right hand side of (4.1), yields

$$\phi_l^{k+1}c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|}\int_{\partial T_l}\vec{F}(c_l^{k+1-\theta})\cdot\vec{n}_l \,\mathrm{d}S = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|}\sum_{j=1}^{4}\int_{e_{jl}}\vec{F}(c_l^{k+1-\theta})\cdot\vec{n}_{jl}\,\mathrm{d}S \quad (4.2)$$

Approximation to the four integrals (advective fluxes) is given by introducing the numerical fluxes H_{jl}^G:

$$\phi_l^{k+1}c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|}\sum_{j=1}^{4} H_{jl}^G \quad (4.3)$$

where H_{jl}^G is the two-point Lipschitz monotone flux (the Godunov flux) depending on the cell averaged values of the concentration variable evaluated on the right side and on the left side of the face e_j, $j = 1, \ldots, 4$ of T_l at the time $t^{k+1-\theta}$ (in the following we omit the time index for sake of clarity). Therefore, H_{jl}^G can be defined as:

$$H_{jl}^G = H(c_{Rj}, c_{Lj}, n_{jl})\overline{T}_{jl} \quad (4.4)$$

where c_{Rj} and c_{Lj} are the reconstruction of c on the right and left side of the face e_j of cell T_l and $\overline{T}_{jl}$ is the surface area of face e_j. High order reconstruction is obtained when the reconstruction is not given by piecewise constant functions (the classical Godunov scheme) but uses a piecewise linear reconstruction together with a limitation procedure.

Using formula (3.8) and applying the components of the normal to the surface $\vec{n}$ described in terms of α, β and γ in Section 3 the components of the advective fluxes along each face become:

$$\int_{e_i}\vec{F}(c)\cdot\vec{n}_{jl}\,\mathrm{d}S = \frac{1}{2}(F(c)_x\alpha + F(c)_y\beta + F(c)_z\gamma)$$

Computation of the values of α, β, γ by formulae described in 3.1.1 does not take into account the sign of the normal, and we are interested to an outward normal. To this aim,

we consider the inner product between the normal and the vector passing through the barycentrum of the tetrahedron and a vertex of the face. If the inner product is positive, then the normal is external, otherwise we are considering an inner normal and change all the sign of α, β, γ. The above investigation is obviously carried out before the advective flux is computed. In the following, we denote as $\alpha_j, \beta_j, \gamma_j$, the components of the normal of the face e_j.

More in detail $\vec{F}(c) = \vec{v}c$ where $\vec{v}$ represents the velocity field. Using the midpoint rule for integrals, the value of concentration on e_j is approximated with the value on its centroid, c_{jl}, and the previous integral is approximated as:

$$\frac{1}{2}c_{jl}(v_x\alpha_j + v_y\beta_j + v_z\gamma_j)$$

The numerical flux H^G_{jl} takes into account the sign of the integral

$$\frac{1}{2}(v_x\alpha_j + v_y\beta_j + v_z\gamma_j)$$

(in practice we look at the sign of the inner product $\vec{v}_{jl} \cdot \vec{n}_{jl}$). If this quantity is positive, we consider the reconstruction at the right c_{Rl}, otherwise c_{Ll} is chosen. In this way the numerical flux H^G_{jl} is given by

$$H^G_{jl} = \frac{1}{2}c_{Xl}(v_x\alpha_j + v_y\beta_j + v_z\gamma_j) \approx \frac{1}{2}c_{jl}(v_x\alpha_j + v_y\beta_j + v_z\gamma_j) \approx \int_{e_j} \vec{F}(c) \cdot \vec{n}_{jl} \, \mathrm{d}S$$

where c_{Xl} may be c_{Rl} or c_{Ll} depending on the choice carried out.

To obtain second order approximation in space, c_{Rj} and c_{Lj} are reconstructed component-wise from the cell averaged data in the following form:

$$c_{Xj} = c_l + \vec{r}_{jl} \cdot \vec{\nabla}(L_l) \qquad X = R, L \tag{4.5}$$

where $\vec{r}_{jl}$ is the vector from the centroid of cell T_l to the centroid of the face e_j and $\vec{\nabla}(L_l)$ is the gradient of a linear reconstruction L_l of c on T_l. L_l is also called the gradient operator.

Selecting L_l to be the gradient operator in (4.5) leads to a second order accurate method (therefore a linear solution is modelled exactly) but doesn't prohibit overshoots and undershoots at the centroids of the tetrahedra faces. Therefore non-linear correction factors called 'limiters' have to be introduced in order to satisfy a local maximum principle.

Usually a limited scheme can be expressed quite simply in two stages as:

1. construct one or more candidates for the linear reconstruction

2. limit the gradient operator chosen from the candidates.

The first step defines a finite set of possible directions of the reconstructed gradient, and the second step chooses one of the directions and bounds the magnitude of the slope.

4.1 The Durlofsky scheme

Extension to tetrahedra of the method developed in [5] is quite easy. In each tetrahedron T_l, four candidates for L_l, designed as L_l^j, $j = 1, 2, 3, 4$ are generated. Indeed, we consider the four neighboring tetrahedra, e.g. T_p, T_q, T_r, T_s, with centroids $\vec{P}_p$, $,\vec{P}_q$, $\vec{P}_r$, $\vec{P}_s$ and construct four linear interpolants. The first candidate L_l^1 interpolates

$$(\vec{P}_l, c_l) \quad (\vec{P}_p, c_p) \quad (\vec{P}_q, c_q) \quad (\vec{P}_r, c_r)$$

L_l^2 is the interpolation of

$$(\vec{P}_l, c_l) \quad (\vec{P}_q, c_q) \quad (\vec{P}_r, c_r) \quad (\vec{P}_s, c_s)$$

L_l^3 and L_l^4 interpolate

$$(\vec{P}_l, c_l) \quad (\vec{P}_r, c_r) \quad (\vec{P}_s, c_s) \quad (\vec{P}_p, c_p)$$

and

$$(\vec{P}_l, c_l) \quad (\vec{P}_s, c_s) \quad (\vec{P}_p, c_p) \quad (\vec{P}_q, c_q)$$

respectively. If a face, say e_i, of T_l is a boundary face, we use $(\vec{P}_{il}, c(\vec{P}_{il}, t))$ instead of - say - $(\vec{P}_p, c_p)$ to build the linear interpolation, where $\vec{P}_{il}$ is the centroid of the face e_i.

We compute the magnitude of the gradient of each L_l^j, putting:

$$|L_l^j| = \sqrt{(\frac{\partial L_l^j}{\partial x})^2 + \frac{\partial L_l^j}{\partial y})^2 + \frac{\partial L_l^j}{\partial z})^2} \quad j = 1, 2, 3, 4.$$

The following procedure for the choice of L_l is used

Algorithm 4.1 (Durlofsky procedure).

1. Select the L_l^j for which $|L_l^j|$ is maximum

2. *Verify the following conditions*

$$\begin{aligned} &L_l^j(\vec{P}_{lp}) \text{ is between } c_l \text{ and } c_p \\ &L_l^j(\vec{P}_{lq}) \text{ is between } c_l \text{ and } c_q \\ &L_l^j(\vec{P}_{lr}) \text{ is between } c_l \text{ and } c_r \\ &L_l^j(\vec{P}_{ls}) \text{ is between } c_l \text{ and } c_s \end{aligned} \tag{4.6}$$

where $\vec{P}_{lp}$ denotes the centroid of the face in common with the T_p tetrahedron. If all conditions (4.6) are satisfied then the procedure is finished and the L_l^j is the appropriate L_l.

3. *If the L_l^j above results in overshoots or undershoots at any one of the four midpoint (that is the above conditions are not satisfied), then select the L_l^j for which $|L_l^j|$ is the second largest and test conditions (4.6) again. If this L_l^i does not satisfy (4.6), select the candidate for which $|L_l^j|$ is the third largest and repeat the procedure. Again, if (4.6) are not satisfied select the interpolant for which $|L_l^j|$ is the minimum.*

4. *If no L_l^j satisfies the requirements (4.6), then choose $L_l = c_l$, that is first order reconstruction.*

Note that the original algorithm [5] does not include the last step of the above description. On the other hand, Liu develops this scheme in two dimensions in the following way [8]: if step 2 of algorithm 4.1 is not verified for any linear interpolant, a local upper and lower bounds are computed taking into account the triangles having at least a common point with the reference triangle. In this way, other conditions should be satisfied, starting from the linear interpolant for which $|L_l^j|$ is the maximum to the candidate with minimum gradient. If these new conditions are not satisfied, the Liu scheme operates with the step (4) described in algorithm 4.1. In three dimensions, the Liu scheme appears too much complicated in the computation of the local upper and lower bounds by considering the nearby edges (the equivalent of the nearby points in two dimensions), that is the tetrahedra having an edge in common with the reference tetrahedron. But Liu's choice carried out in step 4 when conditions (4.6) are not satisfied appears suitable to avoid overshoots and undershoots. For this reason it is added in the algorithm 4.1.

4.2 The *Min* limiter scheme

This method is described in [5] as a procedure analogous to the *min* limiter in second-order ENO (essentially non-oscillatory) schemes [6]. This corresponds to the selection of the L_l^j for which $|L_l^j|$ is minimum. At extrema, that is when c_l is an extremum relative to the values of c at the centroids of the neighbouring tetrahedra, c_p, c_q, c_r and c_s, a first-order approximation is used.

4.3 The average scheme

To verify the effects due to the limiter in the Durlofsky and Min limiter schemes, the reconstruction is calculated as average of the four candidates for the construction of L_l. Therefore it is high order accurate but may produce oscillatory approximation when discontinuities are presents. Indeed, using the previous notation, L_l is set as

$$L_l = \frac{L_l^1 + L_l^2 + L_l^3 + L_l^4}{4}$$

4.4 The Limited Central Difference scheme

The Limited Central Difference (LCD) scheme is presented in [7] for unstructured triangular meshes. Its extension to three dimensions can be described in the following way. Construction of the linear interpolant requires the concentration values of the centroids of the four neighbouring tetrahedra of the reference one. Let L_l^5 be the interpolant of the four pairs of point and corresponding value

$$(\vec{P}_p, c_p) \quad (\vec{P}_q, c_q) \quad (\vec{P}_r, c_r) \quad (\vec{P}_s, c_s)$$

If the reference tetrahedron has any boundary face, we operate as described previously in the construction of the Durlofsky method.

The LCD scheme limits L_l^5 by setting

$$\alpha^v = \begin{cases} \dfrac{\max(c_v - c_l, 0)}{\vec{r}_{vl} \cdot \vec{\nabla}(L_l^5)} & \text{if } \vec{r}_{vl} \cdot \vec{\nabla}(L_l^5) > \max(c_v - c_l, 0) \\ \dfrac{\min(c_v - c_l, 0)}{\vec{r}_{vl} \cdot \vec{\nabla}(L_l^5)} & \text{if } \vec{r}_{vl} \cdot \vec{\nabla}(L_l^5) > \min(c_v - c_l, 0) \qquad v = p, q, r, s \\ 1 & otherwise \end{cases}$$

where $\vec{r}_{vl}$ is the vector from the centroid of T_l to the centroid of the face between cells T_l and T_v and calculating the LCD gradient operator as

$$L_l = \alpha L_l^5 = \min_{v=p,q,r,s} \alpha^v L_l^5.$$

4.5 The Maximum Limited Gradient scheme

The Maximum Limited Gradient (MLG) scheme is proposed in [2] and can be described as a combination of the methodologies proposed for the Durlofsky and LCD schemes.

It takes all five linear interpolants (the candidates for the reconstruction of the Durlofsky scheme and the L_l^5 operator), limits each one in turn in the manner of the LCD scheme and then takes L_l to be the remaining operator with largest gradient.

4.6 The Least Square method with extremum limiter

This method has been proposed directly for tetrahedra in [15]. where Favre-filtered compressible Navier-Stokes equations on a three dimensional unstructured grid of tetrahedral cells are solved.

The reconstruction L_l is computed by minimizing the functional

$$S(L_l) = (L_l(\vec{P}_l) - c_l)^2 + \sum_v (L_l(\vec{P}_v) - c_v)^2 \tag{4.7}$$

where $v = p, q, r, s$.

The minimization is performed by the method of the Least Squares. Then, the approximation of the centroid of each face of T_l is calculated as

$$c_{vl}^{old} = c_l + \vec{r}_{vl} \cdot \vec{\nabla}(L_l) \qquad v = p, q, r, s$$

and the extremum limiter is formulated as Φ

$$\Phi = \begin{cases} 0 & \text{if } \min(c_l, c_v) \leq c_{vl}^{old} \leq \max(c_l, c_v) \\ \max(c_l, c_v) - c_{vl}^{old} & \text{if } c_{vl}^{old} > \max(c_l, c_v) \\ \min(c_l, c_v) - c_{vl}^{old} & \text{if } c_{vl}^{old} < \min(c_l, c_v) \end{cases}$$

The limited value of c_{vl} is given by:

$$c_{vl} = c_{vl}^{old} + \Phi \qquad v = p, q, r, s.$$

The extremum limiter gets its name from the fact that it does not limit the gradients, but directly limits the values of the variables on both sides of the face.

4.7 The Least Square method with the Barth-Jespersen limiter

The gradient operator is obtained minimizing the functional (4.7) by the method of the least squares and is limited by applying the Barth-Jespersen slope limiter [1].

We set

$$c_{vl}^{old} = c_l + \vec{r}_{vl} \cdot \vec{\nabla}(L_l) \qquad v = p, q, r, s \tag{4.8}$$

and compute

$$c_{vl} = c_l + \Phi_l \vec{r}_{vl} \cdot \vec{\nabla}(L_l) \qquad v = p, q, r, s \tag{4.9}$$

where $\Phi_l = \min(\Phi_{l1}, \Phi_{l2}, \Phi_{l3}, \Phi_{l4})$ and

$$\Phi_{jl} = \begin{cases} 1 & \text{if } c_{vl}^{old} - c_l = 0, \\ \min\left(1, \dfrac{c_l^{max} - c_l}{c_{vl}^{old} - c_l}\right) & \text{if } c_{vl}^{old} - c_l > 0, \\ \max\left(1, \dfrac{c_l^{min} - c_l}{c_{vl}^{old} - c_l}\right) & \text{if } c_{vl}^{old} - c_l < 0. \end{cases} \tag{4.10}$$

In the previous formula, $c_l^{min} = \min(c_l, \min_v^4 c_v)$ and $c_l^{max} = \max(c_l, \max_x c_x)$. The application of the slope limiter guarantees that a local maximum principle condition is satisfied.

4.8 Time integration

The time integration is accomplished via a first order explicit Euler scheme or a second order procedure (we choose between midpoint rule in time, Heun scheme and Runge scheme).

The midpoint rule in time is applied when the FV scheme is used in combination with the MHFE scheme in the time splitting technique.

Observe that, in this case, the time-splitting technique introduces an error proportional to the time step and the overall scheme is only first order accurate if no special care is considered in the definition of the numerical flux approximation. Expanding in Taylor series the linear reconstruction of c_l at time $t^{k+1/2}$ is approximated as $L_l^{k+1/2} = L_l^k + (\Delta t/2)\partial c_l^k/\partial t$. Second order of accuracy is obtained if $\partial c_l^k/\partial t$ is evaluated taking into account not only the advective flux but also the dispersive one. This can be implemented by means of backward

finite differencing of the time derivative using the values of concentration at the previous time t^{k-1}, obtaining $L_l^{k+1/2} = L_l^k + (c_l^k - c_l^{k-1})/2$. With this correction term, second order accuracy in time is recovered as proved in [10] for two-dimensional triangular grids. The theorem proved for triangles can be easily extended to tetrahedra.

5 Numerical results

In this Section we report several numerical results and convergence rates obtained in the simulation of simple:

- advection equations - to test and compare the different HRFV schemes described above
- dispersion equations - to test the extension of the MHFE method to tetrahedra
- advection-dispersion problems - applying the more suitable HRFV scheme with the MHFE algorithm in the time splitting technique, to verify that the time splitting technique works well also on tetraedral meshes and can be used to solve more complicated problems.

In combination with advection and advection-dispersion equations, varying values of CFL are considered. When testing the latter equations, also Peclet (Pe) numbers are calculated.

The CFL number is associated to the stability of the FV scheme and is defined for each tetrahedron T_l as

$$\mathrm{CFL} = \Delta t \sup \frac{\overline{T_l}}{|T_l|} \sup |\frac{d\vec{F}}{dc}|$$

where $\overline{T_l} = \sum_{j=1}^{4} \overline{T}_{jl}$ is the total surface area of T_l.

The Peclet number represents the ratio between the advective and the dispersive term and can be defined in our case as [14]

$$\gamma = |D| \Delta t \sup \frac{1}{|T_l|}$$

where $|D|$ is the norm of tensor D. Low Peclet numbers indicate that dispersion is predominant over advection, and vice versa.

About time integration, we employ the Heun scheme for advection problems, the Euler and Crank-Nicolson scheme for dispersion equations, while, for advection-dispersion equations, we use the first order accurate in time and second order in space version of the time-splitting algorithm using a number n_a (≥ 1) of advective time steps per dispersive time step, and the second order accurate in time and space version of the time-splitting algorithm showing that introduction of the correction term in the advection step is crucial to preserve accuracy in time.

We consider one-dimensional and two-dimensional problems solved in three-dimensional grid systems.

More in detail, for one-dimensional problems we consider the domain $[0,1] \times [0,0.1] \times [0,0.1]$ discretized into three different grid levels. The coarsest level ($\ell = 1$) is obtained by uniformly subdividing the domain in triangular prisms that are further subdivided into tetrahedra, obtaining a triangulation with 480 tetrahedra, 189 nodes, and 1128 faces. Uniform subdivision of the prisms yields the next finer meshes, with the finest level formed by 30720 tetrahedra, 6561 nodes, and 64128 faces.

For two-dimensional problems the domain is $[0,1] \times [0,1] \times [0,1]$ and is discretized into three different grid levels, applying the same grid generation procedure employed for the one-dimensional test cases. At the coarsest level ($\ell = 1$) we have eight subdivisions along the y-axis and eight along the z-axis, defining a set of triangular prisms that are again further subdivided into three tetrahedral cells. The coarsest level ($\ell = 1$) is characterized by 3072 tetrahedra, 729 nodes and 6528 faces, the second level by 24576 tetrahedra, 4913 nodes and 50688 faces, while the finest level by 196608 tetrahedra, 35937 nodes and 399360 faces.

Note that in the case of costant coefficients of advection-dispersion equations, Pe decreases by a factor of 4 in passing from a coarser to a finer level.

The numerical convergence of the scheme is evaluated by calculating L_2 relative errors at different grid levels ($|e_{\ell,2}|$) and evaluating the rates of convergence at each level.

The L_2 error norm is calculated using the following formula:

$$|e_{\ell,2}| = \frac{\sqrt{\sum_{l=1}^{m} (c(\vec{P}_l, t^k) - c_l^k)^2}}{\sqrt{\sum_{l=1}^{m} c(\vec{P}_l, t^k)^2}}, \tag{5.1}$$

where $c(\vec{P}_l, t^k)$ is the analytical solution on the centroid of T_l at time t^k and c_l^k is the corresponding numerical solution.

5.1 Advection equations

To compare the accuracy and robusteness of the proposed HRFV schemes, it is sufficient to test simple advection equations whose exact solutions are: $x - v_x t$, $\sin 2\pi(x - v_x t)$, and $\sin 2\pi(x - v_x t) \sin 2\pi(y - v_y t)$. To this aim, appropriate Dirichlet boundary conditions are imposed. Velocity is set equal to $\vec{v} = (1, 0, 0)$ for the one-dimensional tests and equal to $\vec{v} = (0.5, 0.5, 0)$ for the two-dimensional example. We will refer to the above experiments as TEST1, TEST2 and TEST3, respectively.

In addition to the algorithms proposed in Section 4, we also consider the schemes in which no limiter is applied to the interpolant introduced by the LCD method (the L_l^5 operator) and to the interpolant derived by the least squares minimization, the methods obtained limiting the L_l^5 operator by applying the extremum and the Barth-Jespersen limiters respectively, and the method obtained applying the limiter proposed in the LCD scheme to the operator of the least squares minimization. In this way, we can observe the effects of the limiters on different operators.

Therefore we will consider the following schemes:

- Durlofsky method (DUR)
- min limiter method (MIN)
- average method (AVG)
- limited central difference method (LCD)
- no limited central difference method (NLCD)

- limited central difference method with extremum limiter (LCD_EXTR)
- limited central difference method with Barth-Jespersen limiter (LCD_BJ)
- maximum limited gradient scheme (MLG)
- least square method with extremum limiter (LSM_EXTR)
- least square method with Barth-Jespersen limiter (LSM_BJ)
- least square method with LCD limiter (LSM_LCD)
- no limited least square method (NL_LSM)

A time step $\Delta t = 1 \times 10^{-3}$ is used at the coarsest mesh of the one-dimensional tests and halved in passing to the next finer level. The CFL is kept constant at 0.29. In the two-dimensional example the time step is $\Delta t = 4 \times 10^{-3}$ at the coarsest mesh (and halved in passing to the next finer level), and the CFL is equal to 0.33.

Table 5.1 shows the results obtained for the three tests relative to a final time $t^k = 0.1$ s (TEST1 and TEST3) and $t^k = 1$ s (TEST2).

It appears that DUR, LCD, MLG and LSM_LCD display first order of accuracy. An explanation of this order reduction is due to the effect of the limiter applied, leading to the conclusion that HRFV methods working well in two dimensional grids are not readily generalized to three dimension preserving high spatial accuracy. AVG, NLCD, NL_LSM, where no limiter is applied, give the expected convergence rates, confirming that the limiters do not work well in the previous methods. The MIN method give satisfactory results for the one-dimensional examples but display first order accuracy in TEST3. LCD_EXTR and LSM_EXTR are of second order in TEST1 and TEST2 and reach about 1.40 convergence rate in TEST3. Note that the results obtained with both methods are very similar, therefore the limiter effects are dominant to respect the choice of the gradient operator. Finally, LCD_BJ and LSM_BJ are of second order in the one-dimensional tests and achieve about 1.80 order accurate in TEST3 (about 1.85 is the rate observed when no limiter is applied). As before, note that there is no relevant difference between the two methods. Displayed in Figure 5.1 is a log-log plot of L_2 error obtained for TEST3 versus the spacing

between adiacent nodes of the different meshes (theoretical first and second order errors are also plotted to better compare the differents convergence rates).

From the numerical results we can conclude that extension to three dimensions of limiter techniques working satisfactorily in two-dimensions is not so readily as suggested by many authors. The use of linear reconstructions does not always imply second order accuracy even when theoretical considerations seem to suggest high order reconstruction. Three dimensional reconstruction requires careful attention when working on tetrahedral meshes. First order of accuracy achieved by limited schemes may be due to the poorly structured tetrahedra that need to be used in a three dimensional mesh. The Barth-Jespersen limiter seems to overcome this effects and appears as the more efficient between those proposed.

Some open question still remain about the anisotropy of the tetrahedral mesh and the capability of the above limiters of properly correcting the linear reconstruction along all the faces of each tetrahedron. This topic will be discussed in future research.

When applying the time splitting technique, we will operate with the LSM_BJ scheme.

5.2 Dispersion equations

As first test problem we consider the one-dimensional problem whose exact solution is given by $c = 2\exp(-\pi^2 Dt)\sin(\pi x)$. The dispersion coefficient D is set equal to $D = 1 \times 10^{-1}$. As initial time step is used the value $\Delta t_d = 1 \times 10^{-2}$. Convergence results of the MHFE method in combination with the implicit Euler (Eu) or Mid-point (MP) discretization in time are reported in Table 5.2 at the times $t^k = 0.1$ and $t^k = 1$ s. Like predicted from theory, first order of convergence is observed when using the Eu scheme and second order when using the MP scheme.

As two dimensional example we consider the steady-state problem associated to the exact solution $c = \sin(\pi x)\sin(\pi y)$. This test is useful to ascertain superconvergence at the centroids of the tetrahedra. Indeed, the results confirm the theory, as displayed in Table 5.3.

TABLE 5.1: *Advection tests.*

method	ℓ	TEST1		TEST2				TEST3	
		0.1 s		0.1 s		1 s		0.1 s	
		$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate
DUR	1	9.50e-3		3.84e-2		1.28e-1		1.29e-1	
	2	5.09e-3	0.90	1.90e-2	1.01	6.41e-2	1.00	6.56e-2	0.97
	3	2.62e-3	0.96	9.57e-3	0.99	3.25e-2	0.98	3.49e-2	0.91
MIN	1	3.45e-3		1.37e-2		3.66e-2		8.58e-2	
	2	7.98e-4	2.11	4.35e-3	1.65	1.36e-2	1.43	3.48e-2	1.30
	3	2.06e-4	1.95	1.27e-3	1.78	5.13e-3	1.41	2.03e-2	0.78
AVG	1	3.90e-3		1.16e-2		2.26e-2		7.38e-2	
	2	1.01e-3	1.95	3.51e-3	1.72	7.52e-3	1.59	2.42e-2	1.61
	3	2.60e-4	1.96	9.76e-4	1.85	2.18e-3	1.79	6.69e-3	1.85
LCD	1	8.05e-3		3.96e-2		1.47e-1		1.18e-1	
	2	4.91e-3	0.71	1.91e-2	1.05	6.61e-2	1.15	5.67e-2	1.06
	3	2.61e-3	0.91	9.45e-3	1.01	3.12e-2	1.08	2.70e-2	1.07
NLCD	1	3.77e-3		1.12e-2		2.15e-2		7.33e-2	
	2	9.83e-4	1.94	3.33e-3	1.75	7.04e-3	1.61	2.42e-2	1.60
	3	2.50e-4	1.97	9.17e-4	1.86	2.03e-3	1.79	6.67e-3	1.86
LCD_EXTR	1	4.14e-3		1.28e-2		2.27e-2		8.61e-2	
	2	1.04e-3	1.99	3.68e-3	1.80	7.76e-3	1.55	3.30e-2	1.38
	3	2.60e-4	2.00	1.01e-3	1.86	2.37e-3	1.71	1.24e-2	1.41
LCD_BJ	1	3.99e-3		1.24e-2		2.32e-2		7.13e-2	
	2	1.02e-3	1.97	3.63e-3	1.77	7.80e-3	1.57	2.48e-2	1.52
	3	2.58e-4	1.98	1.01e-3	1.84	2.36e-3	1.72	7.24e-3	1.78
MLG	1	1.11e-2		3.61e-2		9.24e-2		1.10e-1	
	2	5.20e-3	1.09	1.74e-2	1.05	4.83e-2	0.93	5.28e-2	1.06
	3	2.61e-3	0.99	8.79e-3	0.98	2.49e-2	0.95	2.74e-2	0.95
LSM_EXTR	1	4.14e-3		1.28e-2		2.25e-2		8.60e-2	
	2	1.03e-3	2.01	3.66e-3	1.81	7.66e-3	1.55	3.31e-2	1.38
	3	2.57e-4	2.00	1.00e-3	1.87	2.33e-3	1.72	1.24e-2	1.42
LSM_BJ	1	3.96e-3		1.24e-2		2.29e-2		7.12e-2	
	2	1.01e-3	1.97	3.60e-3	1.78	7.67e-3	1.58	2.49e-2	1.51
	3	2.55e-4	1.98	1.00e-3	1.85	2.32e-3	1.72	7.24e-3	1.78
LSM_LCD	1	8.05e-3		3.96e-2		1.47e-1		1.18e-1	
	2	4.91e-3	0.71	1.91e-2	1.05	6.61e-2	1.15	5.67e-2	1.06
	3	2.61e-3	0.91	9.45e-3	1.01	3.12e-2	1.08	2.70e-2	1.07
NL_LSM	1	3.72e-3		1.11e-2		2.12e-2		7.32e-2	
	2	9.74e-4	1.93	3.28e-3	1.76	6.91e-3	1.62	2.42e-2	1.60
	3	2.46e-4	1.98	9.02e-4	1.86	1.99e-3	1.79	6.67e-3	1.86

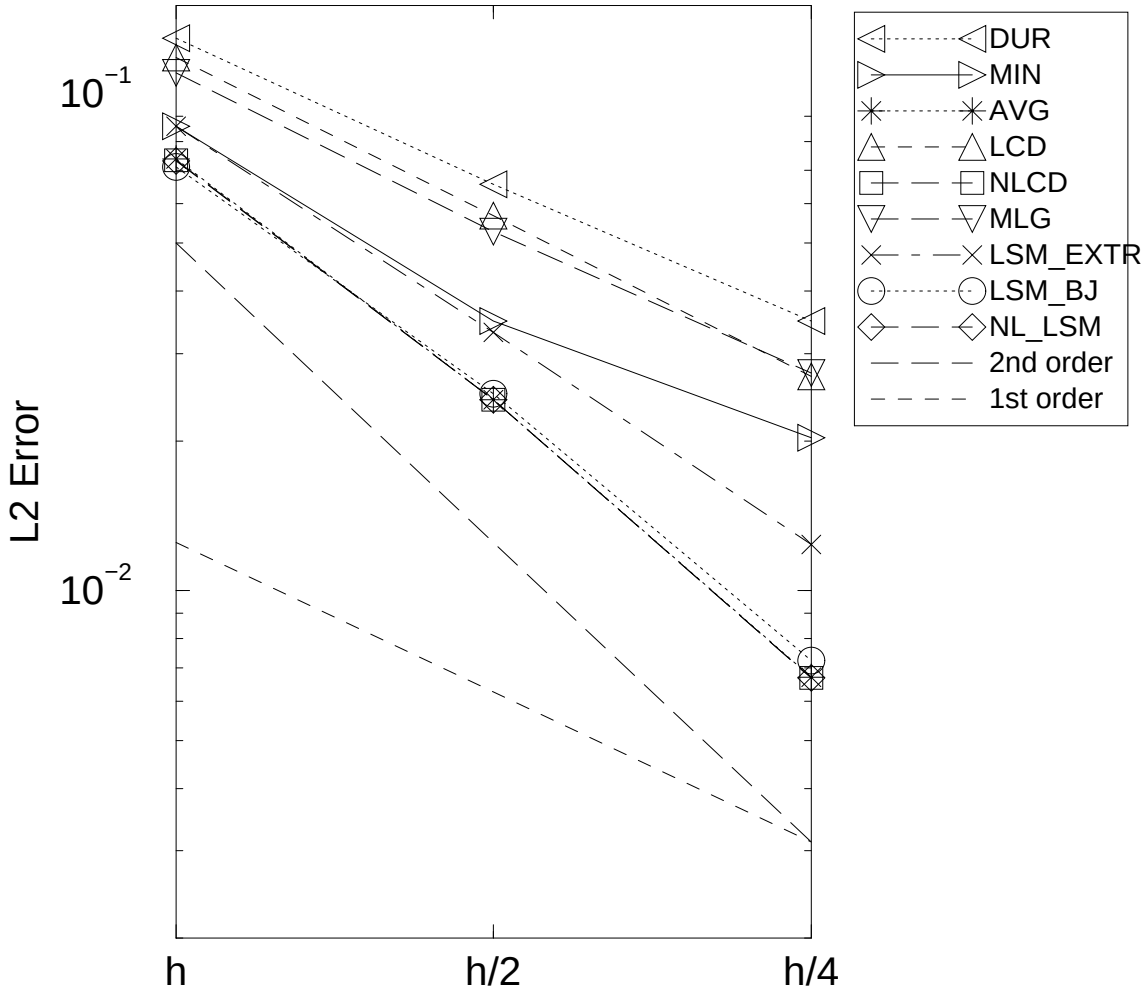

FIGURE 5.1: *Advection TEST3: L_2 error vs spacing between adiacent nodes of the different meshes.*

TABLE 5.2: *Dispersion test. Example 1D.*

ℓ	Eu				MP			
	0.1 s		1 s		0.1 s		1 s	
	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate
1	5.56e-4		5.57e-3		1.96e-4		7.62e-4	
2	2.60e-4	1.10	2.60e-3	1.10	4.88e-5	2.00	1.80e-4	2.08
3	1.26e-4	1.04	1.26e-3	1.04	1.17e-5	2.06	4.38e-5	2.04

TABLE 5.3: *Dispersion test. Example 2D. Steady-state.*

ℓ	$\lvert e_{\ell,2}\rvert$	rate
1	3.17e-2	
2	8.04e-3	1.98
3	2.02e-4	1.99

TABLE 5.4: *Advection-dispersion test. 1D Example 1:* $D = 1.e-2$, $\vec{v} = (1, 0, 0)$.

		Eu-Eu $n_a = 1$		Eu-Eu $n_a = 4$		Eu-Eu $n_a = 8$		MP/NCORR		MP/CORR	
ℓ	Pe	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate
1	0.6	5.78e-3		5.63e-3		5.56e-3		5.87e-3		5.84e-3	
2	0.15	1.32e-3	2.13	1.28e-3	2.14	1.32e-3	2.07	1.35e-3	2.12	1.13e-3	2.37
3	0.038	5.18e-4	1.35	5.20e-4	1.30	5.44e-4	1.28	5.30e-4	1.35	2.64e-4	2.10

5.3 Advection-dispersion equations

First we consider one-dimensional problems solved in a three-dimensional grid system. In the case of costant coefficients, Pe decreases by a factor of 4 in passing from a coarser to a finer level, while CFL is kept constant at 0.29.

5.3.1 Example 1.

The first test solves the transport equation on the rectangular domain previously described, with velocity defined as $\vec{v} = (1, 0, 0)$. The Dirichlet boundary conditions and the source term are determined so that the exact solution is given by $c(\vec{P}, t) = \sin 2\pi(x - t)$. A time step $\Delta t_a = 1 \times 10^{-3}$ is used at the coarsest mesh and halved in passing to the next finer level. The dispersive time step is set equal to the advective one when considering second

TABLE 5.5: *Advection-dispersion equation. 1D Example 1:* $D = 1.e-4$, $\vec{v} = (1, 0, 0)$.

		Eu-Eu $n_a = 1$		Eu-Eu $n_a = 4$		Eu-Eu $n_a = 8$		MP/NCORR		MP/CORR	
ℓ	Pe	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate
1	60	1.11e-2		1.11e-2		1.11e-2		1.15e-2		1.14e-2	
2	15	3.48e-3	1.67	3.47e-3	1.68	3.46e-3	1.68	3.70e-3	1.64	3.44e-3	1.73
3	3.8	1.10e-3	1.66	1.10e-3	1.66	1.09e-3	1.67	1.23e-3	1.59	9.64e-4	1.83

order discretization scheme in time while in the other cases it depends on the number of advective steps n_a allowed for each dispersive time step, following the formula described in Algorithm (2.1). All the results reported are relative to a final time $t^k = 0.1$ s. We compare the time splitting technique with first order of accuracy in time (Eu-Eu) and different advective time steps for dispersive steps (n_a=1,4,8) and applying second order discretization scheme in time adding the correction term in the FV scheme (MP-CORR) and with no correction (MP-NCORR).

Two different values of dimensionless dispersion are considered: $D = 1 \times 10^{-2}$ and $D = 1 \times 10^{-4}$. In the first case dispersion is dominant and Pe varies from 0.6 ($\ell = 1$) to 0.038 ($\ell = 3$). The convergence of the scheme is mainly driven by the MHFE technique. Indeed, with Eu-Eu procedure, the best results are obtained with the same time steps for advection and dispersion ($n_a = 1$) as shown in Table 5.4, while the correction term in the second order time-splitting algorithm is necessary to preserve second order convergence rate. With $D = 1 \times 10^{-4}$ advection is dominant: Pe varies from 60 ($\ell = 1$) to 3.8 ($\ell = 3$) and the advective terms become important, expecting better accuracy for Eu-Eu when $n_a > 1$. Table 5.5 shows the errors and convergence rates for $D = 1 \times 10^{-4}$. We can observe that the differences between the Eu-Eu schemes with $n_a = 1$, $n_a = 4$ and $n_a = 8$ are not important, so that $n_a > 1$ advective steps for dispersive step can be employed, reducing the total number of time steps (and of MHFE linear systems solution) by n_a with significant saving of CPU time and, at the same time, no visible accuracy reduction. Once again we observe that the results obtained with MP-CORR show second order accuracy while the MP-NCORR scheme displays an order of accuracy less than that achieved by the Eu-Eu scheme, according to the theory.

5.3.2 Example 2.

A second test problem considers the partial differential equation describing the movement of a tracer in a semi-infinite column and simulate it on the same three dimensional domain used in the previous example, with $\vec{v} = (1, 0, 0)$ and $D = 1 \times 10^{-2}$. The boundary conditions are $c = 1$ in $x = 0$ and $c = 0$ for $x = \infty$. Zero concentration is used as initial condition. The infinite domain is simulated numerically by employing the grid of unitary length and

TABLE 5.6: *Advection-dispersion equation. 1D Example 2:* $D = 1.e-2$, $\vec{v} = (1, 0, 0)$

		Eu-Eu $n_a = 1$		Eu-Eu $n_a = 4$		Eu-Eu $n_a = 8$		MP/NCORR		MP/CORR	
ℓ	Pe	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate
1	0.6	1.70e-2		1.94e-2		2.34e-2		1.55e-2		1.73e-2	
2	0.15	6.83e-3	1.31	9.38e-3	1.05	1.29e-2	0.86	6.02e-3	1.36	4.83e-3	1.84
3	0.038	3.01e-3	1.18	4.81e-3	0.96	7.56e-3	0.77	2.65e-3	1.18	1.28e-3	1.91

making sure that at the time at which the relative error is evaluated the solution vanishes naturally at the right boundary. The analytical solution to this problem is given by [3]:

$$c(\vec{P}, t) = \frac{1}{2}(\text{erfc}\frac{x - v_1 t}{2\sqrt{Dt}} + \exp\frac{x}{D} \cdot \text{erfc}\frac{x + v_1 t}{2\sqrt{Dt}})$$

The same values of CFL and time step sizes as used previously are employed for this simulation, while Pe varies from 0.6 ($\ell = 1$) to 0.038 ($\ell = 3$). Table 5.6 shows the errors and convergence rates at the different levels confirming the considerations stated above for the same dispersion and velocity parameters.

5.3.3 Two-dimensional test

We now describe a two-dimensional problem solved in the three-dimensional grid system. We consider a velocity field $\vec{v} = (.5, .5, 0)$ and vary the size of the dispersion coefficient. The Dirichlet boundary conditions and the source term are determined so that the exact solution is given by $c(\vec{P}, t) = \sin 2\pi(x - v_1 t) \sin 2\pi(y - v_2 t)$.

Table 5.7 reports the results to the problem with $D = 4 \times 10^{-4}$ at time $t^k = 0.1$ s. The CFL number for this test case is set equal to 0.33 and Pe varies between 67 ($\ell = 1$) to 4.2 ($\ell = 3$). Advection is dominant and the HRFV scheme error contributes the most to the truncation error. In this case, the differences between MP/CORR and MP/NCORR are not important and larger dispersion time step sizes could be used without loss of accuracy. An efficient choice in this case could be for example the Eu/Eu scheme with $n_a = 8$.

TABLE 5.7: *Advection-dispersion equation. 2D Example:* $D = 4.e-4$, $\vec{v} = (0.5, 0.5, 0)$

		Eu-Eu $n_a = 1$		Eu-Eu $n_a = 4$		Eu-Eu $n_a = 8$		MP/NCORR		MP/CORR	
ℓ	Pe	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate	$\lvert e_{\ell,2}\rvert$	rate
1	67	7.06e-2		7.04e-2		7.00e-2		7.06e-2		6.89e-2	
2	17	2.33e-2	1.60	2.32e-2	1.60	2.29e-2	1.61	2.32e-2	1.60	2.30e-2	1.58
3	4.2	6.18e-3	1.91	6.11e-3	1.92	6.02e-3	1.93	6.17e-3	1.91	6.01e-3	1.94

6 Conclusions

In this paper we have presented the extension on tetrahedra of the time splitting technique developed on triangular meshes. Extension of this technique required careful attention on the adoption of suitable HRFV schemes in order to preserve second order spatial accuracy.

Indeed, generalizing the MHFE method to three dimensions required only some computational effort to calculate integrals on three dimensions. On the other hand, the numerical results confirmed the well-known properties of this scheme (as superconvergence on the centroids of the cells).

On the contrary, among the generalizations of several FV schemes developed on two dimensional unstructured grids, it was difficult to obtain the same degree of robustness and accuracy as in two dimensions. Therefore, some open question still remain about the anisotropy of the tetrahedral mesh and the capability of the limiters of properly correcting the linear reconstruction.

An efficient strategy has been found by applying a second order linear reconstruction using the least squares method with the limiting procedure based on the Barth-Jespersen limiters, as the numerical results for advection problems confirm (the LSM_BJ method).

The time splitting technique developed on tetrahedra has employed the MHFE and the LSM_BJ schemes for discretizing the dispersion and the advection terms, respectively. Simple test problems have confirmed the time splitting technique as an effective tool for the solution of the advection-dispersion equation not only in two dimensional triangular grids but also on tetrahedral meshes.

A Construction of the RT0 basis functions in two dimensions

In two dimensions and working on triangles T_l, with edges e_1, e_2, e_3 and nodes $\vec{P}_1, \vec{P}_2, \vec{P}_3$, (we use a similar notation as well as for tetrahedra), the RT0 basis functions are of the following form:

$$\vec{w}_{il} = \begin{pmatrix} a_{il}x + b_{il} \\ a_{il}y + c_{il} \end{pmatrix} \qquad i = 1, 2, 3.$$

The normal to an edge $e_1 = \vec{P}_1\vec{P}_2$ is derived by the plane:

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0$$

that is $\alpha(x - x_1) + \beta(y - y_1) = 0$, where

$$\alpha = y_2 - y_1, \qquad \beta = x_1 - x_2.$$

The normalized components of the normal to e_1 are then (with no consideration about inner or outward normal) :

$$n_x = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}}, \qquad n_y = \frac{\beta}{\sqrt{\alpha^2 + \beta^2}}.$$

Property (3.9) applied on two dimensions gives the following result:

$$\begin{aligned}
&\int_{e_i} \vec{w}_{jl} \cdot \vec{n}_{il} \, \mathrm{d}S = \int_{e_i} (ax + b)n_x + (ay + c)n_y \, \mathrm{d}S = \\
&a \int_{\vec{P}_1}^{\vec{P}_2} \{xn_x + [y_1 - \frac{n_x}{n_y}(x - x_1)]n_y\} \frac{1}{|n_y|} \, \mathrm{d}x + \int_{\vec{P}_1}^{\vec{P}_2} (bn_x + cn_y) \frac{1}{|n_y|} dx = \\
&a \int_{\vec{P}_1}^{\vec{P}_2} (x_1n_x + y_1n_y) \frac{1}{|n_y|} \, \mathrm{d}x + \int_{\vec{P}_1}^{\vec{P}_2} (bn_x + cn_y) \frac{1}{|n_y|} \, \mathrm{d}x = \\
&a(x_1n_x + y_1n_y) \frac{|\vec{P}_1\vec{P}_2|}{|n_y|} + (bn_x + cn_y) \frac{|\vec{P}_1\vec{P}_2|}{|n_y|} = \delta_{ij}
\end{aligned}$$

where $|\vec{P}_1\vec{P}_2|$ is the length of the edge e_1, that is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{\alpha^2 + \beta^2}$. Therefore, we obtain:

$$\int_{e_i} \vec{w}_{jl} \cdot \vec{n}_{il} \, \mathrm{d}S = (x_1\alpha + y_1\beta) + (b\alpha + c\beta) = \delta_{ij}$$

The 2D classical Galerkin functions $N_i, i = 1, 2, 3$, whose gradients are normal to the edges e_{i+1} for $i = 1, 2$ while the gradient of N_3 is normal to e_1, can be written as:

$$N_i = \frac{a_i + b_i x + c_i y}{2A},$$

where A is the area of the triangle T_l. In particular, the coefficients are:

$$a_1 = \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} \quad a_2 = -\begin{vmatrix} x_1 & y_1 \\ x_3 & y_3 \end{vmatrix} \quad a_3 = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}$$

$$b_1 = \begin{vmatrix} 1 & y_2 \\ 1 & y_3 \end{vmatrix} \quad b_2 = -\begin{vmatrix} 1 & y_1 \\ 1 & y_3 \end{vmatrix} \quad b_3 = \begin{vmatrix} 1 & y_1 \\ 1 & y_2 \end{vmatrix}$$

$$c_1 = -\begin{vmatrix} 1 & x_2 \\ 1 & x_3 \end{vmatrix} \quad c_2 = \begin{vmatrix} 1 & x_1 \\ 1 & x_3 \end{vmatrix} \quad c_3 = -\begin{vmatrix} 1 & x_1 \\ 1 & x_2 \end{vmatrix}$$

It is simple to see that, for example, for the edge e_1 the following relations hold:

$$\alpha x_1 + \beta y_2 = -a_3, \qquad \alpha = b_3, \qquad \beta = c_3$$

Thus

$$\int_{e_1} \vec{w}_j \cdot \vec{n}_1 \, \mathrm{d}S = -a_3 a + (b_3 b + c_3 c). \tag{A.1}$$

The unknowns are now the coefficients a, b, c of $\vec{w}_j$. Setting $j = 1$, from relation (3.9) and substituting (A.1), we get:

$$\begin{aligned} -a_3 a + b_3 b + c_3 c &= 1 \\ -a_1 a + b_1 b + c_1 c &= 0 \\ -a_2 a + b_2 b + c_2 c &= 0 \end{aligned} \tag{A.2}$$

Note that $\sum_{i=1}^{3} b_i = \sum_{i=1}^{3} c_i = 0$ and that $\sum_{i=1}^{3} a_i = 2A$, where A is the area of T_l:

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

Therefore system (A.2) gives the following relationship between a and A:

$$-2Aa = 1 \Leftrightarrow a = -\frac{1}{2A}.$$

Application of divergence theorem to $\int_T \vec{\nabla} \cdot \vec{w}_i \, \mathrm{d}x$ (following the same lines as in the 3D case) assures that the value of a is the same along the three edges of T_l and is positive and equal to $\dfrac{1}{2|A|}$.

Again, considering the first equation of system (A.2) and the measure of A, we obtain for $\vec{w}_1$:

$$b_1 = \frac{-x_3}{2|A|}, \qquad c_1 = \frac{-y_3}{2|A|}.$$

Similarly, the coefficients for $\vec{w}_2$ and $\vec{w}_3$ are:

$$b_j = \frac{-x_{j-1}}{2|A|}, \qquad c_j = \frac{-y_{j-1}}{2|A|}.$$

B About computation of the advective fluxes in 2D

When calculating the advective flux in two dimensions, we take into account that

$$\int_{P_1}^{P_2} \vec{v} \cdot \vec{n} \, \mathrm{d}S = \int v_x n_x + v_y n_y \, \mathrm{d}S \approx (v_x n_x + v_y n_y)|\vec{P_1}\vec{P_2}| = v_x \alpha + v_y \beta$$

We have approximated velocity with the value on the midpoint of the edge e_1.

References

[1] T. J. Barth and D. C. Jespersen, *The design and application of upwind schemes on unstructured meshes*, AIAA paper, 89-0366 (1989), pp. 1–12.

[2] P. Batten, C. Lambert, and D. Causon, *Positively conservative high-resolution convection schemes for unstructured meshes*, Int. J. Numer. Methods Eng., 39 (1996), pp. 1821–1838.

[3] J. Bear, *Hydraulics of Groundwater*, McGraw-Hill, New York, 1979.

[4] C. N. Dawson, *Godunov-mixed methods for advection-diffusion equations in multi-dimensions*, SIAM J. Num. Anal., 30 (1993), pp. 1315–1332.

[5] L. J. Durlofsky, B. Engquist, and S. Osher, *Triangle based adaptive stencils for the solution of hyperbolic conservation laws*, J. Comp. Phys., 98 (1992), pp. 64–73.

[6] A. Harten, S. Osher, B. Engquist, and S. R. Chakravarthy, *Some results on uniformly high-order accurate essentially nonoscillatory schemes*, Appl. Num. Math., 2 (1986), pp. 347–377.

[7] M. E. Hubbard, *Multidimensional slope limiters for MUSCL-type Finite Volume schemes on unstructured grids*, J. Comp. Phys., 155 (1999), pp. 54–74.

[8] X.-D. Liu, *A maximum principle satisfying modification of triangle based adaptive stencils for the solution of scalar hyperbolic conservation laws*, SIAM J. Num. Anal., 30 (1993), pp. 701–716.

[9] A. Mazzia, *Mixed Finite Elements and Finite Volumes for the solution of density dependent flow and transport of radioactive contaminants in porous media*, PhD thesis, Università di Padova, December 1999.

[10] A. Mazzia, L. Bergamaschi, C. N. Dawson, and M. Putti, *Godunov mixed methods on triangular grids for advection-dispersion equations*, Comput. Geosc., 6 (2002), pp. 123–139.

[11] A. Mazzia, L. Bergamaschi, and M. Putti, *A time-splitting technique for advection-dispersion equation in groundwater*, J. Comput. Phys., 157 (2000), pp. 181–198.

[12] ——, *Triangular finite volume-mixed finite element discretization for the advection-diffusion equation*, in Large Scale Scientific Computations of Engineering and Environmental Sciences, M. Griebel, S. Margenov, and P. Yalamov, eds., Braunschweig (Ger), 2000, Vieweg, pp. 371–378.

[13] ——, *A second order time-splitting technique for advection-dispersion equation on unstructured grids*, in Godunov Methods Theory and Applications, E. F. Toro, ed., vol. 1, New York, 2001, Academic/Plenum Publishers, pp. 603–610.

[14] M. Putti, W. W.-G. Yeh, and W. A. Mulder, *A triangular finite volume approach with high resolution upwind terms for the solution of groundwater transport equations*, Water Resour. Res., 26 (1990), pp. 2865–2880.

[15] H. Yan, G. Urbin, D. Knight, and A. A. Zheltovodov, *Compressible large eddy simulation using unstructured grid: supersonic boundary layer and compression ramps*, in 10th Inter. Conference On Methods Of Aerophysical Research, Novosibirsk (Russia), 2000.